Team T

1. Hemanth Thathireddy (A20525346)
2. Satya Dineswara Reddy Satti (A20528639)
3. Sai Manasa Basani (A20544372)
4. Alavalapati Veera manohar Reddy (A20526070)
5. Ying Zhang (A20547843)

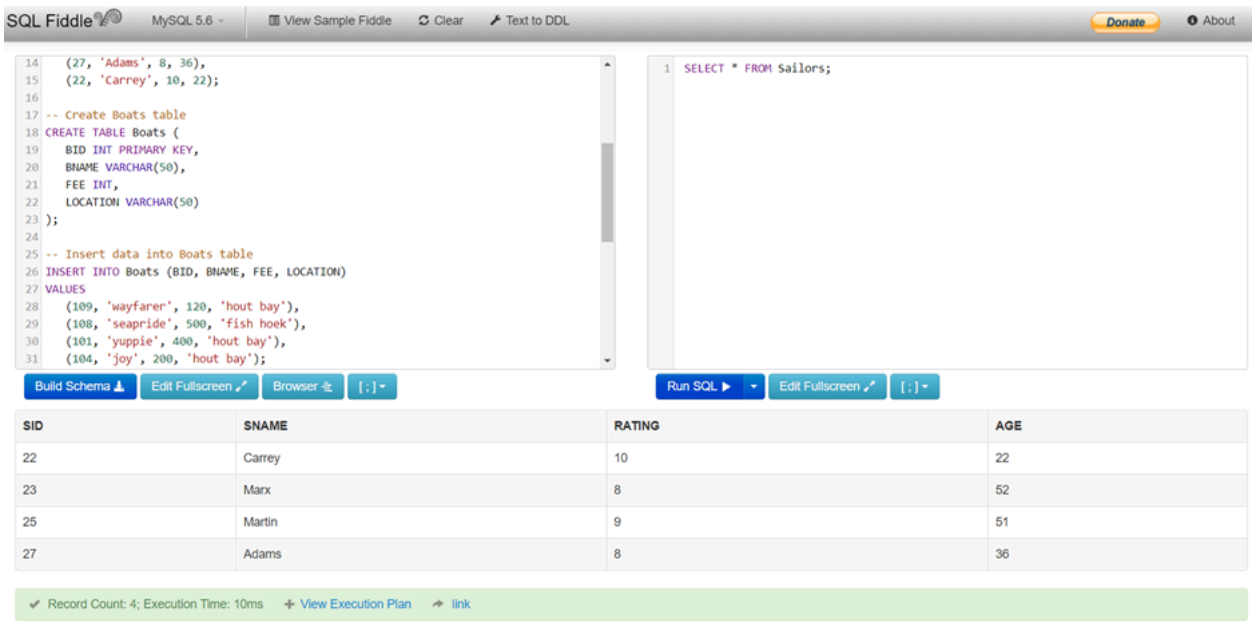
1.

a) Find everything found in the Sailors table

b) SQL Command:

SELECT * FROM Sailors;

c) Result:



2.

a) Get sailor ID, rank & age of all sailors, ordered from highest to lowest rank.

b) SQL Command:

SELECT SID, RATING * 10 AS RANK, AGE FROM Sailors ORDER BY RANK DESC;

c) Result:

```
   (27, 'Adams', 8, 36),
   (22, 'Carrey', 10, 22);

-- Create Boats table
CREATE TABLE Boats (
   BID INT PRIMARY KEY,
   BNAME VARCHAR(50),
   FEE INT,
   LOCATION VARCHAR(50)
);

-- Insert data into Boats table
INSERT INTO Boats (BID, BNAME, FEE, LOCATION)
VALUES
   (109, 'wayfarer', 120, 'hout bay'),
   (108, 'seapride', 500, 'fish hoek'),
   (101, 'yuppie', 400, 'hout bay'),
   (104, 'joy', 200, 'hout bay');
```

Build Schema | Edit Fullscreen | Browser | [;]

```
SELECT SID, RATING * 10 AS RANK, AGE FROM Sailors ORDER BY RANK DESC;
```

Run SQL | Edit Fullscreen | [;]

SID	RANK	AGE
22	100	22
25	90	51
23	80	52
27	80	36

Record Count: 4; Execution Time: 15ms | View Execution Plan | link

Algebra expression: **π(SID, RATING * 10, AGE)σ(RATING)(SAILORS)**

3.

a) Get alphabetical list of sailors with rating less than 10.

b) SQL Command:

SELECT SNAME FROM Sailors WHERE RATING < 10 ORDER BY SNAME;

c) Result:

```
   (27, 'Adams', 8, 36),
   (22, 'Carrey', 10, 22);

-- Create Boats table
CREATE TABLE Boats (
   BID INT PRIMARY KEY,
   BNAME VARCHAR(50),
   FEE INT,
   LOCATION VARCHAR(50)
);

-- Insert data into Boats table
INSERT INTO Boats (BID, BNAME, FEE, LOCATION)
VALUES
   (109, 'wayfarer', 120, 'hout bay'),
   (108, 'seapride', 500, 'fish hoek'),
   (101, 'yuppie', 400, 'hout bay'),
   (104, 'joy', 200, 'hout bay');
```

Build Schema | Edit Fullscreen | Browser | [;]

```
SELECT SNAME FROM Sailors WHERE RATING < 10 ORDER BY SNAME;
```

Run SQL | Edit Fullscreen | [;]

SNAME
Adams
Martin
Marx

Record Count: 3; Execution Time: 8ms | View Execution Plan | link

4.

a) Find how much deposit money there is in total and how many tuples are in the Reserves table.

b) SQL Command:

SELECT SUM(DEPOSIT) AS TOTAL_DEPOSIT, COUNT(*) AS NUM_TUPLES FROM Reserves;

c) Result:

```
14    (27, 'Adams', 8, 36),
15    (22, 'Carrey', 10, 22);
16
17 -- Create Boats table
18 CREATE TABLE Boats (
19    BID INT PRIMARY KEY,
20    BNAME VARCHAR(50),
21    FEE INT,
22    LOCATION VARCHAR(50)
23 );
24
25 -- Insert data into Boats table
26 INSERT INTO Boats (BID, BNAME, FEE, LOCATION)
27 VALUES
28    (109, 'wayfarer', 120, 'hout bay'),
29    (108, 'seapride', 500, 'fish hoek'),
30    (101, 'yuppie', 400, 'hout bay'),
31    (104, 'joy', 200, 'hout bay');
```

```
1  SELECT SUM(DEPOSIT) AS TOTAL_DEPOSIT, COUNT(*) AS NUM_TUPLES FROM Reserves;
```

Build Schema | Edit Fullscreen | Browser | [;] | Run SQL | Edit Fullscreen | [;]

TOTAL_DEPOSIT	NUM_TUPLES
220	4

Record Count: 1; Execution Time: 17ms + View Execution Plan link

5.

a) Get all info on boats in locations similar to "Fishhoek."

b) SQL Command:

SELECT * FROM Boats WHERE LOCATION LIKE '%IS%K';

c) Result:

SQL Fiddle | MySQL 5.6 | View Sample Fiddle | Clear | Text to DDL | Donate | About

```
14    (27, 'Adams', 8, 36),
15    (22, 'Carrey', 10, 22);
16
17 -- Create Boats table
18 CREATE TABLE Boats (
19    BID INT PRIMARY KEY,
20    BNAME VARCHAR(50),
21    FEE INT,
22    LOCATION VARCHAR(50)
23 );
24
25 -- Insert data into Boats table
26 INSERT INTO Boats (BID, BNAME, FEE, LOCATION)
27 VALUES
28    (109, 'wayfarer', 120, 'hout bay'),
29    (108, 'seapride', 500, 'fish hoek'),
30    (101, 'yuppie', 400, 'hout bay'),
31    (104, 'joy', 200, 'hout bay');
```

```
1  SELECT * FROM Boats WHERE LOCATION LIKE '%IS%K';
```

Build Schema | Edit Fullscreen | Browser | [;] | Run SQL | Edit Fullscreen | [;]

BID	BNAME	FEE	LOCATION
108	seapride	500	fish hoek

Record Count: 1; Execution Time: 1ms + View Execution Plan link

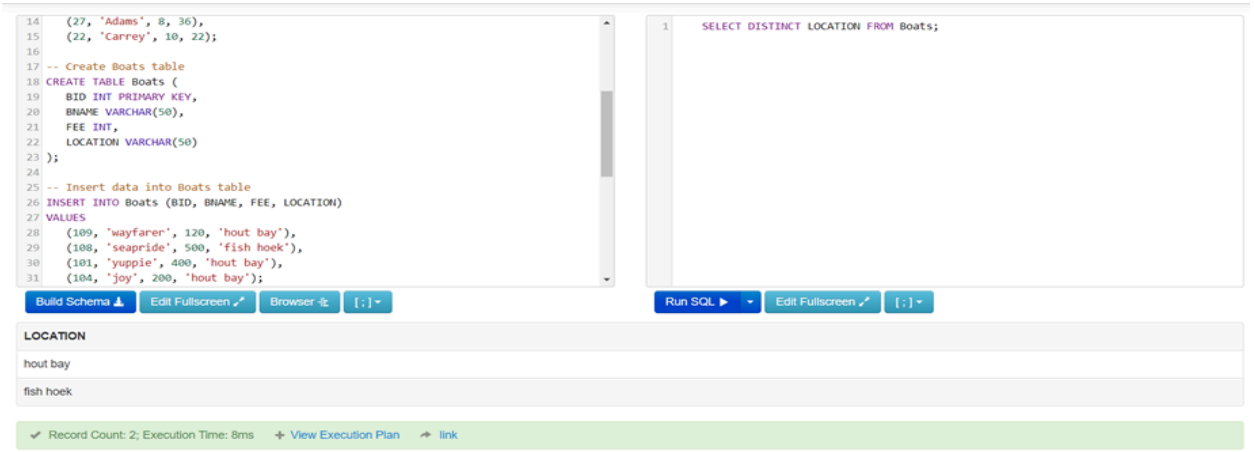
6.

a) Get distinct locations where boats are kept.

b) SQL Command:

SELECT DISTINCT LOCATION FROM Boats;

c) Result:



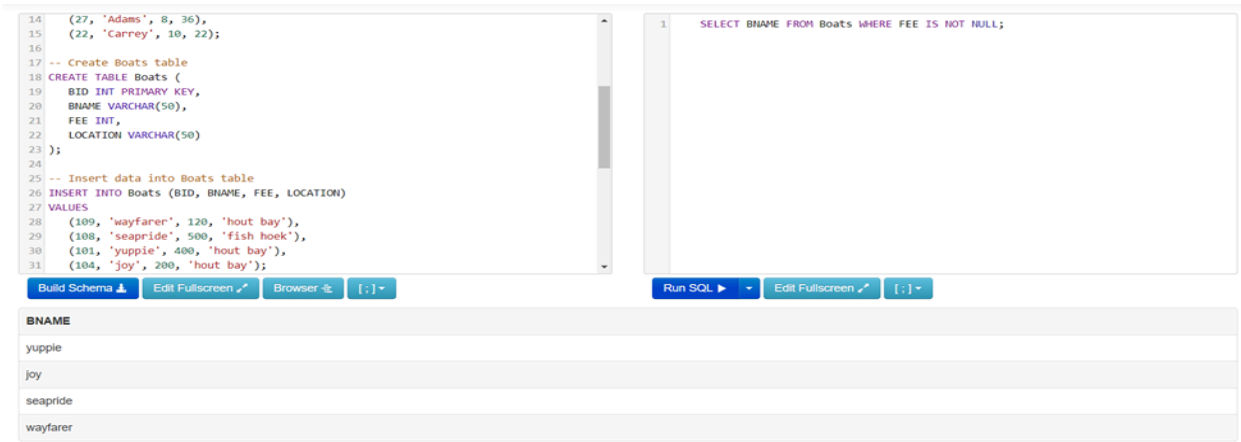
7.

a) Get the names of all boats that have a fee value recorded in the database.

b) SQL Command:

SELECT BNAME FROM Boats WHERE FEE IS NOT NULL;

c) Result:



8.

a) Get ID of all boats that have not been reserved.

b) SQL Command:

SELECT BID FROM Boats WHERE BID NOT IN (SELECT BID FROM Reserves);

c) Result:

```
(27, 'Adams', 8, 36),
(22, 'Carrey', 10, 22);

-- Create Boats table
CREATE TABLE Boats (
   BID INT PRIMARY KEY,
   BNAME VARCHAR(50),
   FEE INT,
   LOCATION VARCHAR(50)
);

-- Insert data into Boats table
INSERT INTO Boats (BID, BNAME, FEE, LOCATION)
VALUES
   (109, 'wayfarer', 120, 'hout bay'),
   (108, 'seapride', 500, 'fish hoek'),
   (101, 'yuppie', 400, 'hout bay'),
   (104, 'joy', 200, 'hout bay');
```

```
SELECT BID FROM Boats WHERE BID NOT IN (SELECT BID FROM Reserves);
```

Build Schema | Edit Fullscreen | Browser | [;] | Run SQL | Edit Fullscreen | [;]

Record Count: 0; Execution Time: 13ms — View Execution Plan — link

id	select_type	table	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	Boats	index		PRIMARY	4		4	100.00	Using where; Using index
2	DEPENDENT SUBQUERY	Reserves	index_subquery	BID	BID	4	func	1	100.00	Using index

Algebra expression: π(BOAT_ID)(BOATS - (σ(BOAT_ID = RESERVES.BOAT_ID)(RESERVES)))

9.

a) Get all reservation info, including all details on the boats being reserved.

b) SQL Command:

SELECT * FROM Reserves INNER JOIN Boats ON Reserves.BID = Boats.BID;

c) Result:

SQL Fiddle | MySQL 5.6 | View Sample Fiddle | Clear | Text to DDL | Donate | About

```
(27, 'Adams', 8, 36),
(22, 'Carrey', 10, 22);

-- Create Boats table
CREATE TABLE Boats (
   BID INT PRIMARY KEY,
   BNAME VARCHAR(50),
   FEE INT,
   LOCATION VARCHAR(50)
);

-- Insert data into Boats table
INSERT INTO Boats (BID, BNAME, FEE, LOCATION)
VALUES
   (109, 'wayfarer', 120, 'hout bay'),
   (108, 'seapride', 500, 'fish hoek'),
   (101, 'yuppie', 400, 'hout bay'),
   (104, 'joy', 200, 'hout bay');
```

```
SELECT * FROM Reserves INNER JOIN Boats ON Reserves.BID = Boats.BID;
```

Build Schema | Edit Fullscreen | Browser | [;] | Run SQL | Edit Fullscreen | [;]

SID	BID	DAY	DEPOSIT	BID	BNAME	FEE	LOCATION
25	101	2023-08-29	50	101	yuppie	400	hout bay
23	104	2023-08-29	0	104	joy	200	hout bay
27	108	2023-08-29	70	108	seapride	500	fish hoek
22	109	2023-08-29	100	109	wayfarer	120	hout bay

Record Count: 4; Execution Time: 10ms — View Execution Plan — link

Algebra expression: ρ(ReservationInfo, (σ(RESERVES.BOAT_ID = BOATS.BOAT_ID)(RESERVES ⋈ BOATS)))

10.

a) For all reservations, get the name of the sailor, along with the day and name of the boat booked.

b) SQL Command:

SELECT S.SNAME, R.DAY, B.BNAME

FROM Sailors AS S

INNER JOIN Reserves AS R ON S.SID = R.SID

INNER JOIN Boats AS B ON R.BID = B.BID;

c) Result:

```
    (27, 'Adams', 8, 36),
    (22, 'Carrey', 10, 22);

-- Create Boats table
CREATE TABLE Boats (
    BID INT PRIMARY KEY,
    BNAME VARCHAR(50),
    FEE INT,
    LOCATION VARCHAR(50)
);

-- Insert data into Boats table
INSERT INTO Boats (BID, BNAME, FEE, LOCATION)
VALUES
    (109, 'wayfarer', 120, 'hout bay'),
    (108, 'seapride', 500, 'fish hoek'),
    (101, 'yuppie', 400, 'hout bay'),
    (104, 'joy', 200, 'hout bay');
```

Build Schema | Edit Fullscreen | Browser | [;]

```
    SELECT S.SNAME, R.DAY, B.BNAME
FROM Sailors AS S
INNER JOIN Reserves AS R ON S.SID = R.SID
INNER JOIN Boats AS B ON R.BID = B.BID;
```

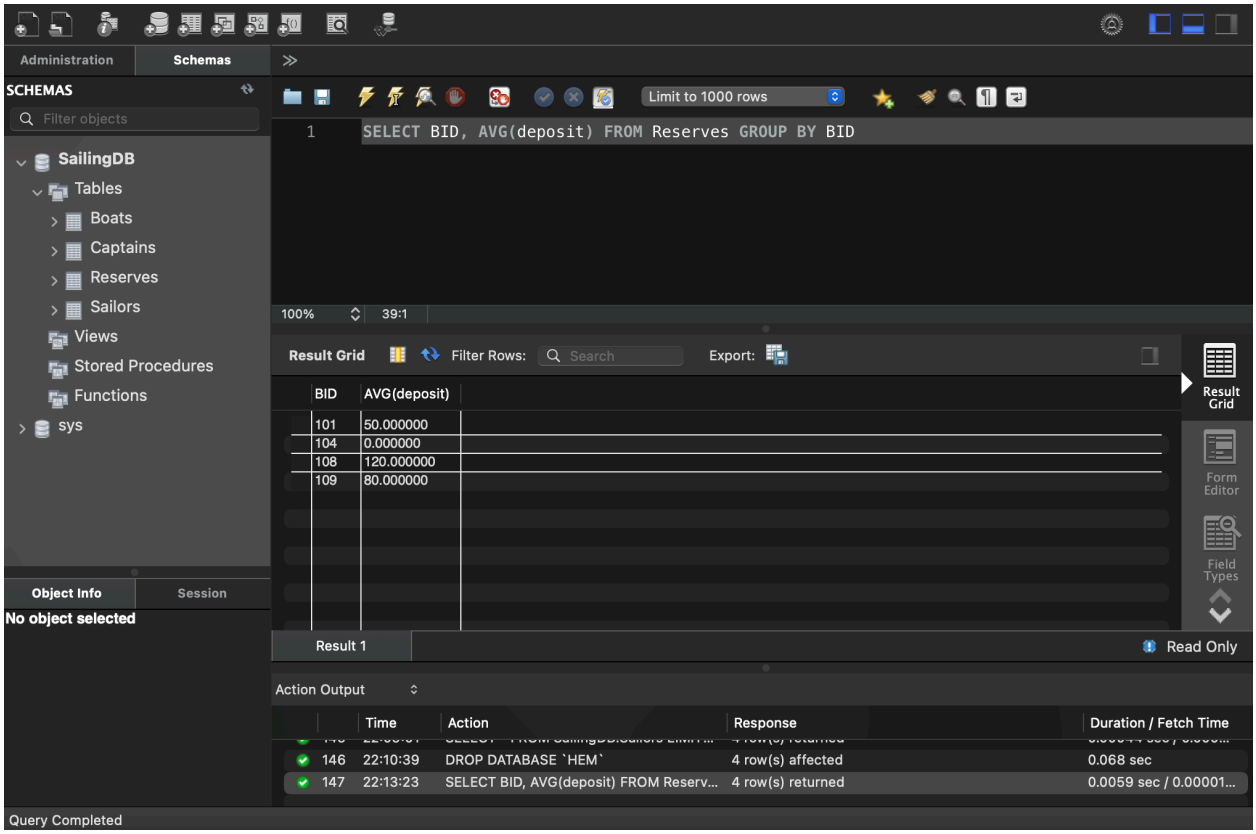
Run SQL | Edit Fullscreen | [;]

SNAME	DAY	BNAME
Carrey	2023-08-29	wayfarer
Marx	2023-08-29	joy
Martin	2023-08-29	yuppie
Adams	2023-08-29	seapride

Algebra expression: ρ(ReservationDetails, (σ(SAILORS.SID = RESERVES.SID)⋈(RESERVES ⋈BOATS)))

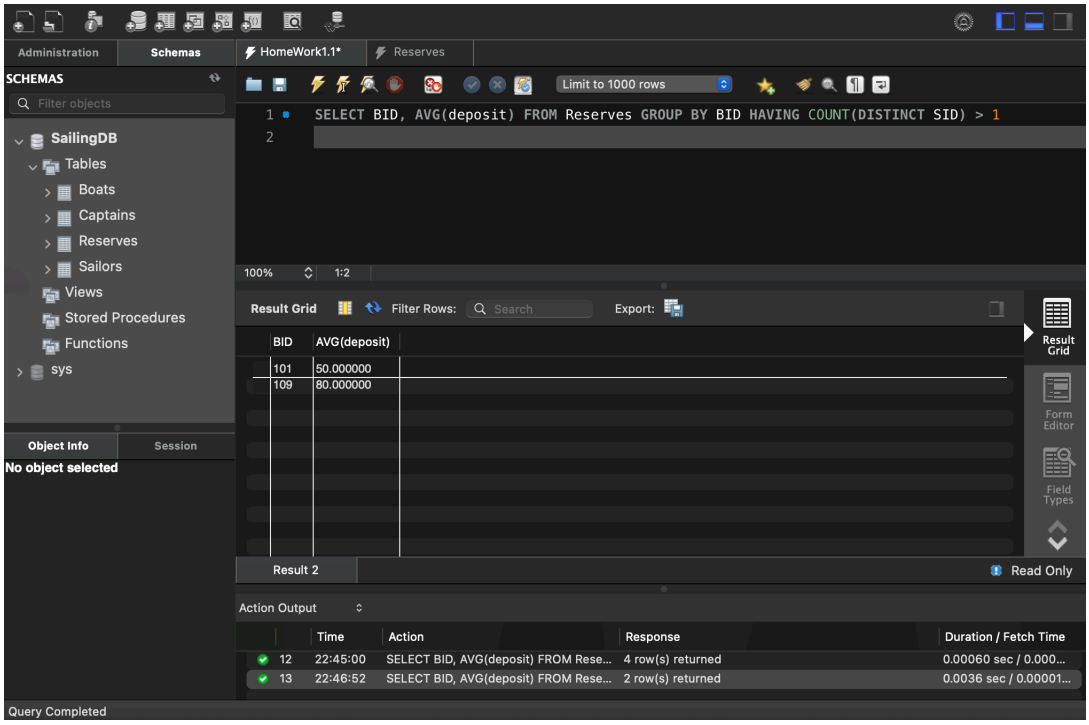
11. Get the average deposit paid for each boat.

SELECT BID, AVG(deposit) FROM Reserves GROUP BY BID



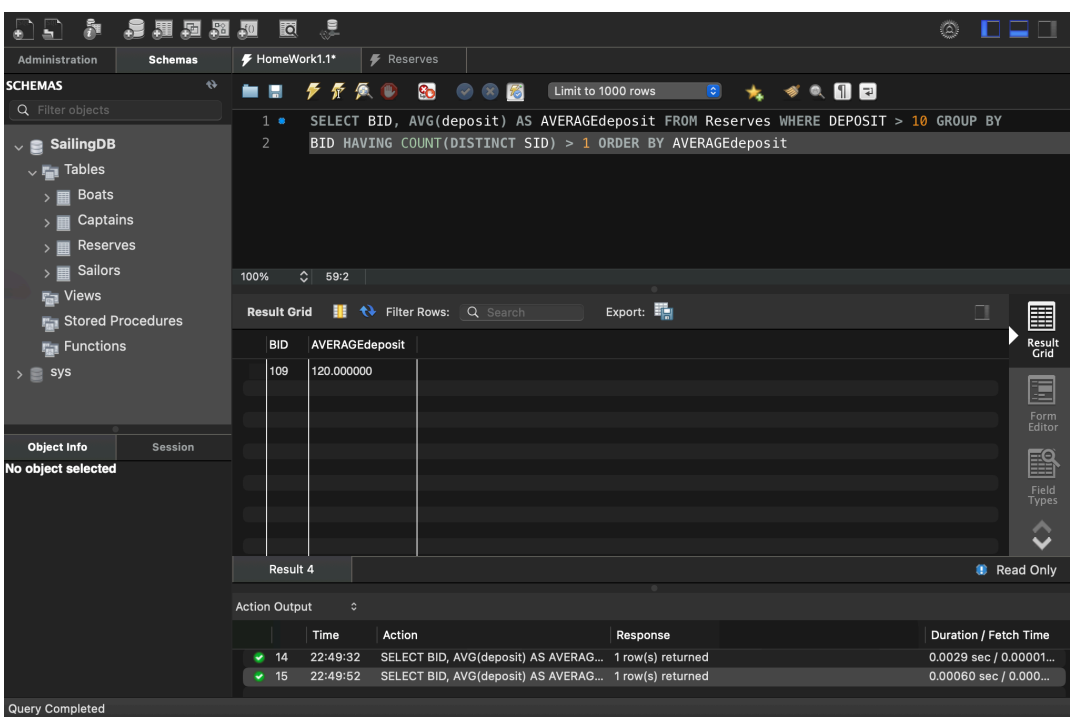
12. Get the average deposit paid for each boat that has been booked by more than one person.

SELECT BID, AVG(deposit) FROM Reserves GROUP BY BID HAVING COUNT(DISTINCT SID) > 1



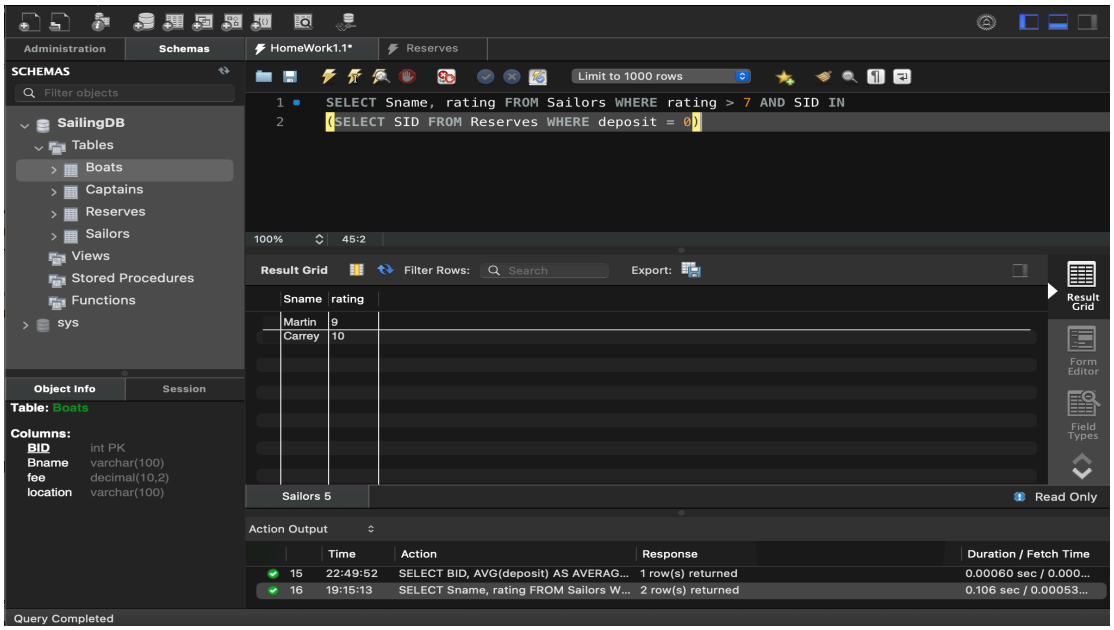
13. Get the average firm deposit paid for each boat that has been booked by more than one person, in increasing order of amount. A firm deposit is one which exceeds R10.

SELECT BID, AVG(deposit) AS AVERAGEdeposit FROM Reserves WHERE DEPOSIT > 10 GROUP BY BID HAVING COUNT(DISTINCT SID) > 1 ORDER BY AVERAGEdeposit



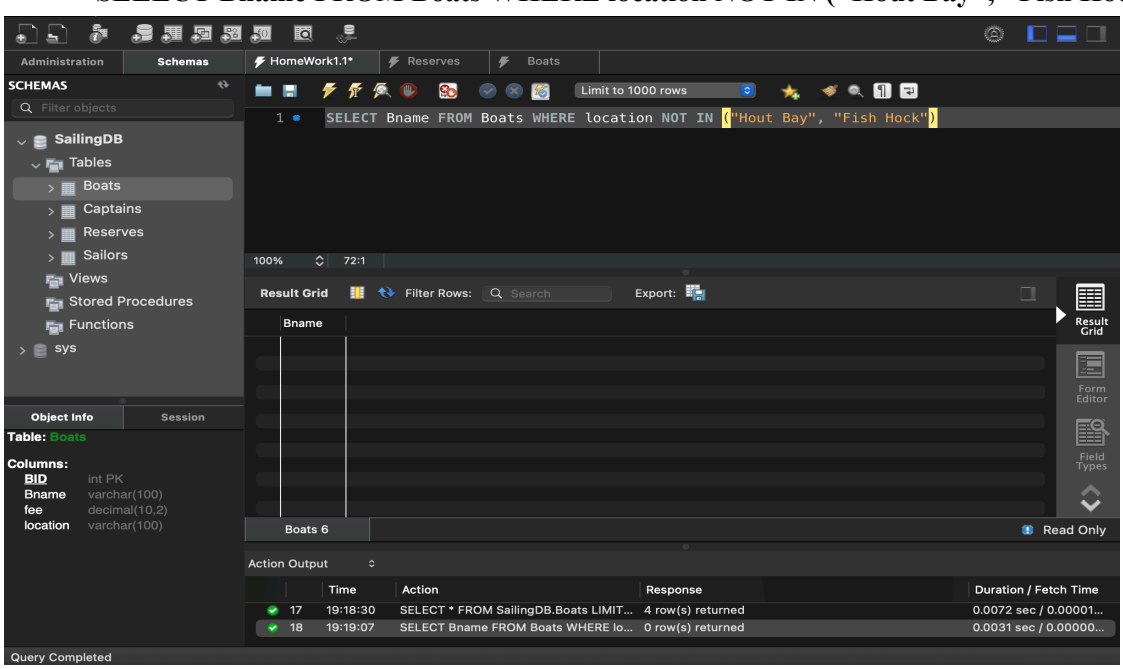
14. Get name & rating of sailors with rating exceeding 7 who made any reservation with 0 deposit.

SELECT Sname, rating FROM Sailors WHERE rating > 7 AND SID IN (SELECT SID FROM Reserves WHERE deposit = 0)



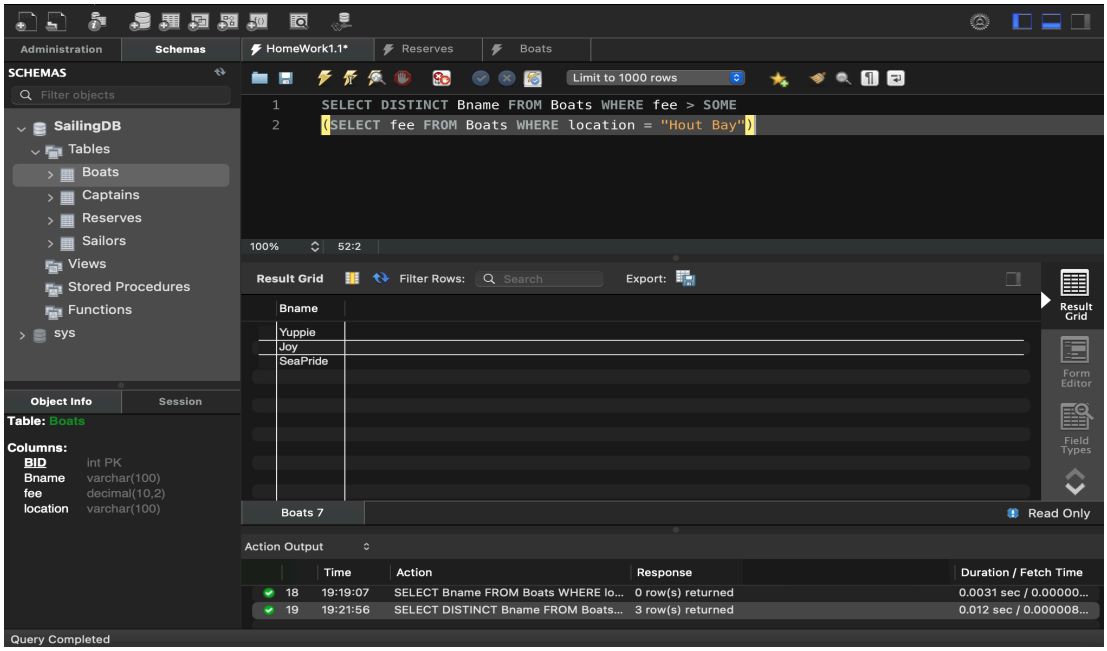
15. Get names of boats located in a place other than Hout Bay or Fish Hoek.

SELECT Bname FROM Boats WHERE location NOT IN ("Hout Bay", "Fish Hock")



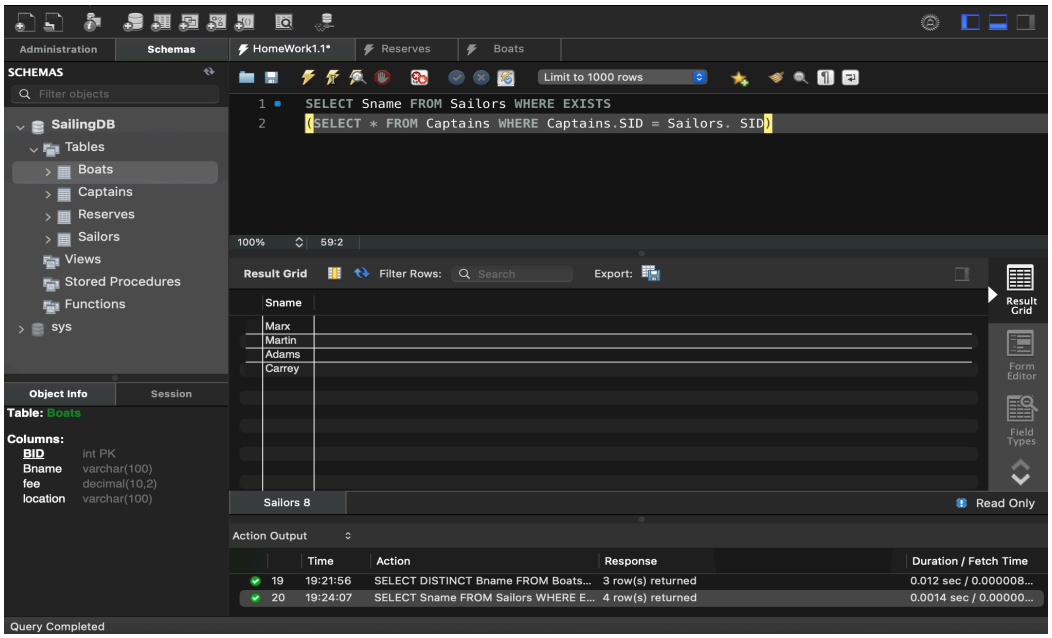
16. Get names of boats having a fee larger than any boat located in Hout Bay.

SELECT DISTINCT Bname FROM Boats WHERE fee > SOME (SELECT fee FROM Boats WHERE location = "Hout Bay")



17. Get names that are in both the sailors and the captains relations

SELECT Sname FROM Sailors WHERE EXISTS (SELECT * FROM Captains WHERE Captains.SID = Sailors. SID)



Algebra expression: **π Sname (Sailors) $\cap$ π Sname (Captains)**

18. Get names of boats that have exactly 1 reservation.

SELECT B.BNAME

FROM Boats AS B

WHERE B.BID IN (

SELECT R.BID

FROM Reserves AS R

GROUP BY R.BID

HAVING COUNT(*) = 1

);

```
   (27, 'Adams', 8, 36),
   (22, 'Carrey', 10, 22);

-- Create Boats table
CREATE TABLE Boats (
   BID INT PRIMARY KEY,
   BNAME VARCHAR(50),
   FEE INT,
   LOCATION VARCHAR(50)
);

-- Insert data into Boats table
INSERT INTO Boats (BID, BNAME, FEE, LOCATION)
VALUES
   (109, 'wayfarer', 120, 'hout bay'),
   (108, 'seapride', 500, 'fish hoek'),
   (101, 'yuppie', 400, 'hout bay'),
   (104, 'joy', 200, 'hout bay');
```

Build Schema | Edit Fullscreen | Browser | [;]

```
 SELECT B.BNAME
FROM Boats AS B
WHERE B.BID IN (
    SELECT R.BID
    FROM Reserves AS R
    GROUP BY R.BID
    HAVING COUNT(*) = 1
);
```

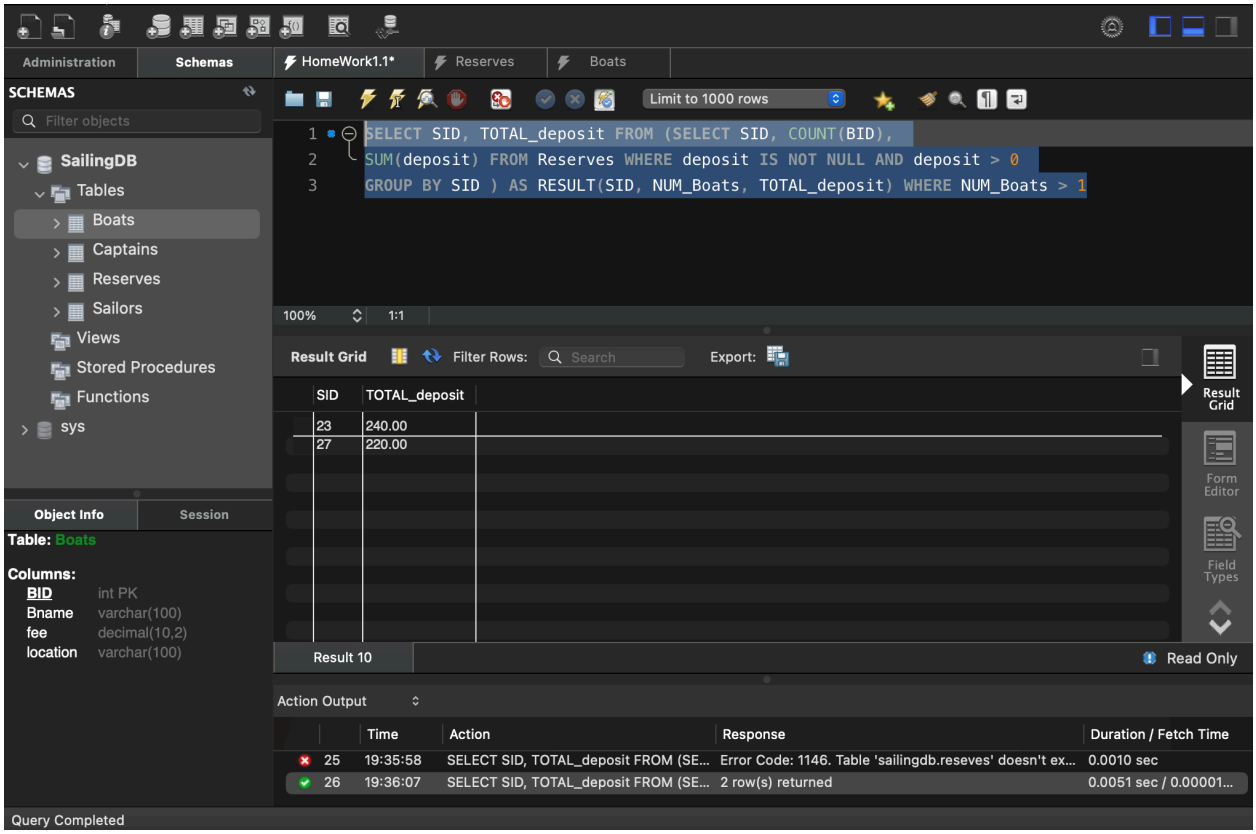
Run SQL | Edit Fullscreen | [;]

BNAME
yuppie
joy
seapride
wayfarer

Record Count: 4; Execution Time: 4ms | View Execution Plan | link

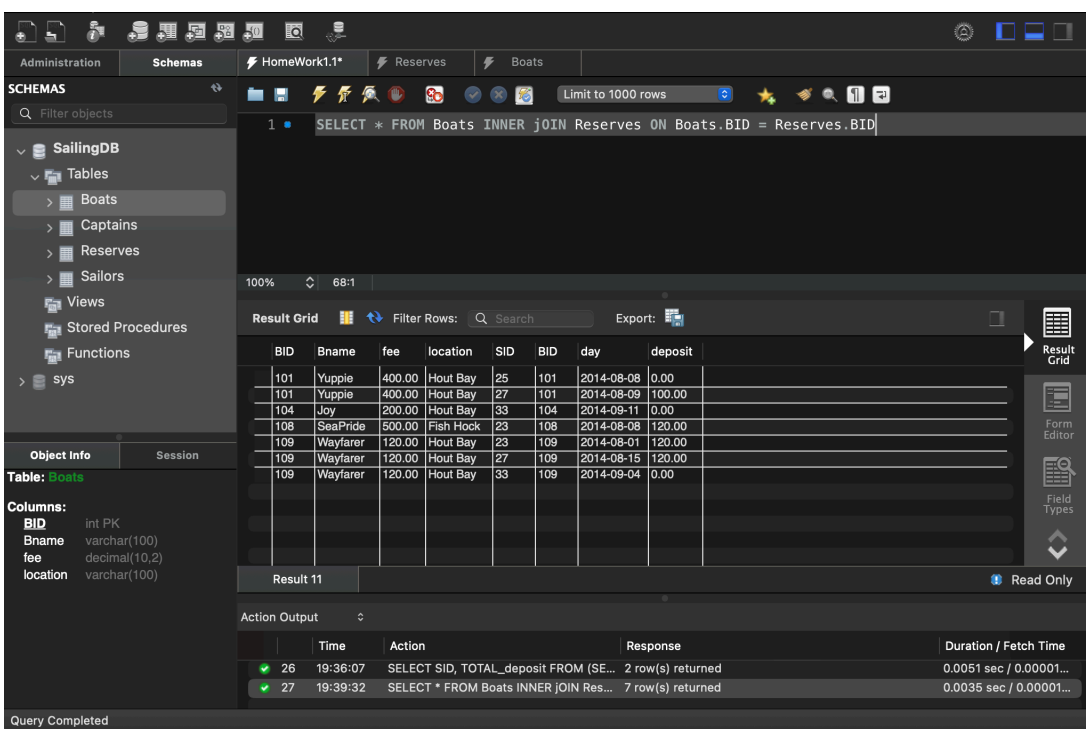
19. Get sailor ID and total deposit paid for sailors who have booked more than 1 boat

SELECT SID, TOTAL_deposit FROM (SELECT SID, COUNT(BID), SUM(deposit) FROM Reserves WHERE deposit IS NOT NULL AND deposit > 0 GROUP BY SID) AS RESULT(SID, NUM_Boats, TOTAL_deposit) WHERE NUM_Boats > 1



20. Get all reservation info including details of the boat booked.

SELECT * FROM Boats INNER jOIN Reserves ON Boats.BID = Reserves.BID



Algebra expression: ρ(ReservationInfo, (σ(Reserves.Boat_ID = Boats.Boat_ID)(Reserves ⋈ Boats)))

21.

a) Get all information on every boat. If a boat has reservations, include all its reservation info.

b) SQL Command:

SELECT * FROM Boats LEFT OUTER JOIN Reserves on Boats.Bid= Reserves.Bid

c) Result:

21 Input

```
SELECT * FROM Boats LEFT OUTER JOIN Reserves on Boats.BID = Reserves.BID
```

Output

BID	Bname	fee	location	SID	BID	day	deposit
109	Wayfarer	120	Hout Bay	23	109	2014-08-01	120
109	Wayfarer	120	Hout Bay	27	109	2014-08-15	120
109	Wayfarer	120	Hout Bay	33	109	2014-09-04	0
108	SeaPride	500	Fish Hock	23	108	2014-08-08	120
101	Yuppie	400	Hout Bay	25	101	2014-08-08	0
101	Yuppie	400	Hout Bay	27	101	2014-08-09	100
104	Joy	200	Hout Bay	33	104	2014-09-11	0

22.

a) Create a new tuple for the boat named “Nino” which has fee R150, BID 110, and is in Fish Hoek.

b) SQL Command:

INSERT INTO Boats (BNAME, BID, FEE, LOCATION)

VALUES ('Nino', 110, 150, 'Fish Hoek');

c) Result:

22 Input

```
INSERT INTO Boats VALUES ("Nino",110,150,"Fish Hoek")
```

Available Tables

Boats

BID	Bname	fee	location
109	Wayfarer	120	Hout Bay
108	SeaPride	500	Fish Hock
101	Yuppie	400	Hout Bay
104	Joy	200	Hout Bay
Nino	110	150	Fish Hoek

23.

a) Remove all bookings from Reserves where there is no deposit.

b) SQL Command:

DELETE FROM Reserves

WHERE DEPOSIT IS NULL OR DEPOSIT = 0;

c) Result:

23 Input

Run SQL

Available Tables

```
DELETE FROM Reserves WHERE deposit is NULL OR deposit = 0
```

Reserves

SID	BID	day	deposit
23	109	2014-08-01	120
23	108	2014-08-08	120
27	101	2014-08-09	100
27	109	2014-08-15	120

24.

a) Increase the fee of every boat by 12%.

b) SQL Command:

UPDATE Boats

SET FEE = FEE * 1.12;

c) Result:

24 Input

Run SQL

Available Tables

```
UPDATE Boats SET fee = fee*1.12
```

Boats

BID	Bname	fee	location
109	Wayfarer	134.4	Hout Bay
108	SeaPride	560	Fish Hock
101	Yuppie	448.00000000000006	Hout Bay
104	Joy	224.00000000000003	Hout Bay
Nino	110	168.00000000000003	Fish Hoek

25.

a) Make a view called Bookings which hides the Deposit value i.e. only has the other 3 attributes.

b) SQL Command:

CREATE VIEW Bookings AS

SELECT SID, BID, DAY

FROM Reserves;

c) Result:

26. Create a table called Reserves with 3 integer attributes BID, SID & deposit, and a date attribute Day. Allow only deposit to be omitted, and ensure SID and BID values exist in the database. When someone books a boat it is for the whole day.

b) SQL Command:

CREATE TABLE Reserves (

BID INTEGER NOT NULL,

SID INTEGER NOT NULL,

DAY DATETIME NOT NULL,

DEPOSIT INTEGER,

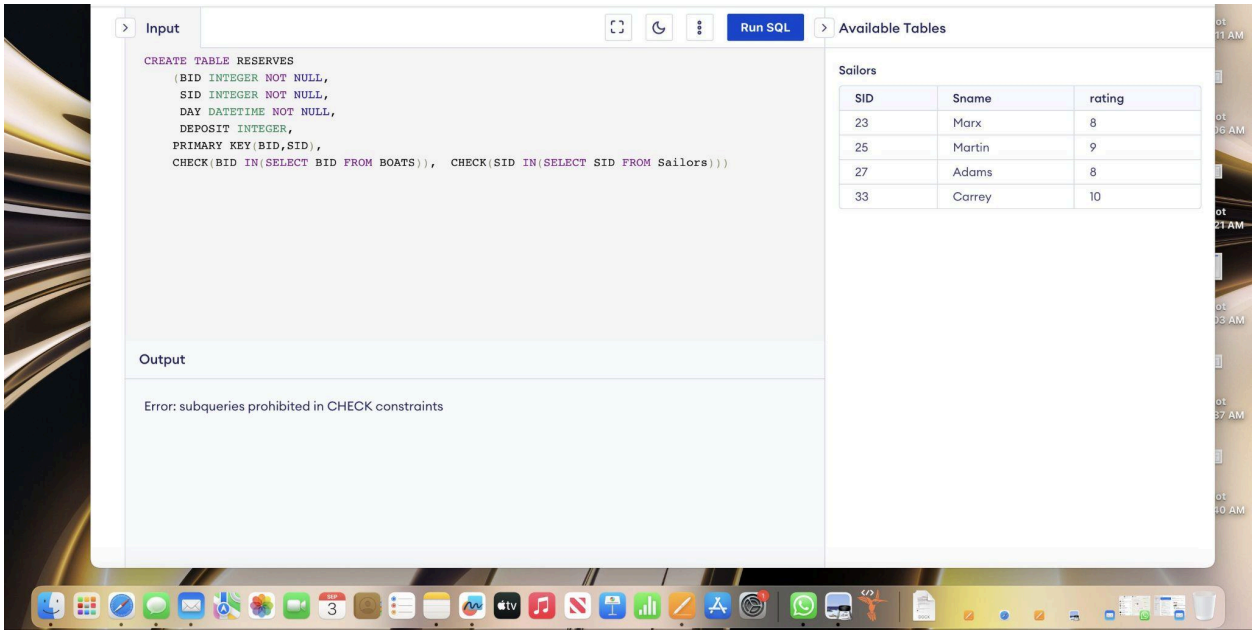
PRIMARY KEY (BID, SID),

CHECK (BID IN (SELECT BID FROM Boats)),

CHECK (SID IN (SELECT SID FROM Sailors))

);

c) Result:



27.

a) Add a new attribute NEEDSREPAIR to the Boats table. It is usually “N”.

b) SQL Command:

ALTER TABLE Boats

ADD NEEDSREPAIR CHAR(1) DEFAULT 'N';

c) Result:

28. We should not be ageist. Remove the Age attribute.

a) Remove the Age attribute from the Sailors table.

b) SQL Command:

ALTER TABLE Sailors DROP AGE;

c) Result:

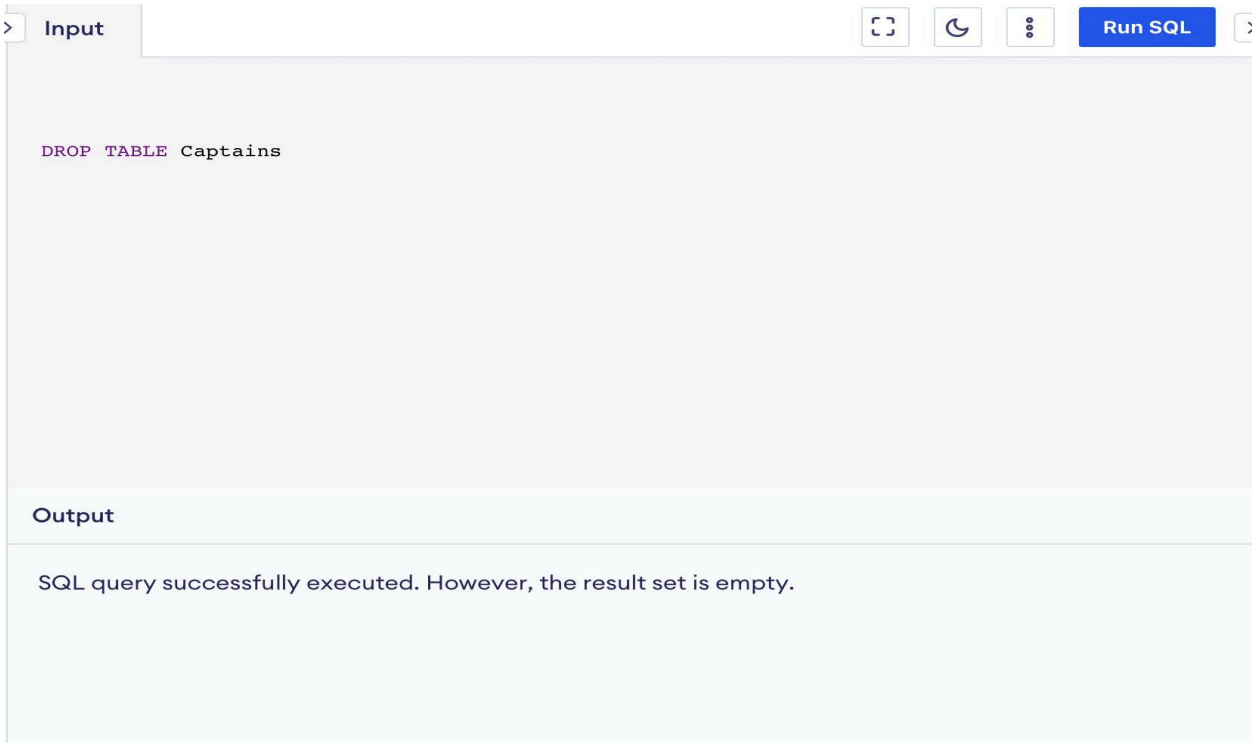
29.

a) Remove the Captains relation altogether so that nobody can try insert or use Captains in future.

b) SQL Command:

DROP TABLE Captains;

c) Result:



Team Contributions:

S.No	Name	Contribution(%)
1.	Hemanth Thathireddy	20%
2.	Satya Dineswara Reddy Setti	20%
3.	Sai Manasa Basani	20%
4.	Alavalapati Veera manohar Reddy	20%
5.	Ying Zhang	20%